

PATENT APPLICATION

**METHOD AND SYSTEM FOR REAL-TIME QUERYING, RETRIEVAL
AND INTEGRATION OF DATA FROM DATABASES OVER A
COMPUTER NETWORK**

Inventor(s): Ey-Chih Chow, a citizen of the United States, residing at
7494 Bollinger Road,
Cupertino, CA 95014

Assignee: Accelerate Software, Inc.
21771 Stevens Creek Boulevard, #200
Cupertino, CA 95014

Entity: Small Business Concern

**METHOD AND SYSTEM FOR REAL-TIME QUERYING, RETRIEVAL
AND INTEGRATION OF DATA FROM DATABASES OVER A
COMPUTER NETWORK**

5 **CROSS-REFERENCES TO RELATED APPLICATION(S)**

[01] The present application claims the benefit of priority under 35 U.S.C. § 119 from U.S. Provisional Patent Application Serial No. 60/273,816, entitled "METHOD AND SYSTEM FOR REAL-TIME QUERYING, RETRIEVAL AND INTEGRATION OF DATA FROM DATABASES OVER A COMPUTER NETWORK" filed on March 6, 2001, the disclosure of which is hereby incorporated by reference in its entirety for all purposes.

BACKGROUND OF THE INVENTION

[02] The present invention generally relates to data retrieval. More specifically, the present invention relates to a method and system for retrieving and integrating data from one or more databases over a computer network.

[03] As business-to-business (B2B) technology becomes more widespread, a number of companies have implemented B2B platforms, along the way defining protocols which allow for automated standardized interactions between some of their business partners. Most often, these protocols are designed to model the paper-based processes - orders, invoices, etc. - in an effort to more efficiently execute such processes and reduce the associated costs. The business objective was to lower operating costs.

[04] The evolution of the Internet as a communication vehicle between businesses has allowed many companies to utilize business-to-business software platforms to link simple business processes and transactions with each other - orders, invoices, etc. However, this has still not enabled businesses within a value chain to truly collaborate and share information to allow business partners to make intelligent decisions about when, where and how to conduct these transactions.

[05] B2B transaction automation, outside the firewall, tracks a similar pattern to the internal transaction automation companies implemented in the 1980s, and the e-commerce transaction automation implemented in the late 1990s. Different technologies were used -- CICS with COBOL for internal business transactions, and e-commerce servers with Java for

e-commerce. The same result occurred. Standard transactions and processes were captured and automated to save execution costs.

[06] Simple transaction execution provides the first level of automation but does not wring all the costs out of the business processes involved. History has shown that once simple transactions were defined, the business problems evolved to require more complex decision-making and intelligence.

[07] Today's computer networking environments and technologies, such as the electronic data interchange (EDI), email and ftp, are conventionally used by enterprises to share information across a business supply chain for forecasting, planning and execution. However, when information must be collected and generated in short periods of time such as on an hourly basis or even real time, these technologies often perform below expectations.

[08] Various systems were introduced in an attempt to improve the above-mentioned situation. For example, one system was introduced to solve planning issues such as retailer forecasting and inventory management by connecting retailers' and suppliers' resources through direct connection of their computers through networking. The forecasting is calculated by doing a series of reviews on an order in a basically single-to-single enterprise basis.

[09] Another example is a system that allows data access outside a single enterprise. The interchange layer of the system allows the system to see the whole supply chain instead of a single enterprise, which is mostly beneficial to supply chain planning. Data from the supply chain is collected and stored on the database. The data is then processed using certain parameters and processes to provide supply information for supply chain planning. Basically, using certain parameters such as manufacturing capacity, ERP and financial support, a forecasting module can be set up to evaluate information that is necessary for supply chain planning. Data is collected and then computed before it is accessible for planning. The system is designed to shorten the time to collect and compute a large variety and amount of data. It is useful for forecasting and planning purpose. However, the system only allows for access of computed data that can be computed with a certain amount of time delay. It cannot provide up-to-date and accurate data to support the supply chain real time decision-making. When a specific piece of data is required by the user that is out of the syllabus of the forecasting model, the computed data also lacks flexibility to fulfill the requirement.

[10] Therefore, it would be desirable to provide a method and system which is capable of querying, retrieving and integrating data from databases over a computer network on a real-time basis in a more efficient manner.

SUMMARY OF THE INVENTION

[11] A method and system for retrieving and integrating data from multiple databases over a computer network is provided. An exemplary embodiment of the present invention includes a system having an aggregation server and a number of agents. The aggregation server is capable of communicating with the agents via a computer network such as the Internet. Each agent is designed to communicate locally with a number of data sources. A user is able to retrieve data from the data sources by contacting the aggregation server which, in turn, causes the appropriate agents to retrieve the requested data from the relevant data sources.

[12] According to an exemplary embodiment, when the user issues a request to the aggregation server to retrieve certain data, the request is converted into an internal query by the aggregation server. The internal query is then matched against a set of rules. Each rule specifies how an internal query is to be partially satisfied using one or more data sources. For a set of rules matching the internal query, a subquery is generated. All the generated subqueries of the internal query are then used by their respective agents to retrieve the requested data. Optionally, all the generated subqueries are optimized to effect more efficient retrieval of data from the respective data sources. When the requested data is returned from all the relevant agents, the requested data is then joined, fused and unioned to produce the final results representing the collective data responsive to the internal query. Star union is used to optimize the union and join operations.

[13] Reference to the remaining portions of the specification, including the drawings and claims, will realize other features and advantages of the present invention. Further features and advantages of the present invention, as well as the structure and operation of various embodiments of the present invention, are described in detail below with respect to accompanying drawings, like reference numbers indicate identical or functionally similar elements.

BRIEF DESCRIPTION OF THE DRAWINGS

[14] Fig. 1 is a simplified block diagram illustrating an exemplary embodiment of the present invention;

[15] Fig. 2 is a flow diagram illustrating the data integration process performed by an exemplary embodiment of the present invention;

[16] Fig. 3 is an illustrative example of an input query request in accordance with an exemplary embodiment of the present invention;

[17] Fig. 4 is an illustrative example of a query definition file in accordance with an exemplary embodiment of the present invention;

[18] Fig. 5 is an illustrative example of a mediator specification file in accordance with an exemplary embodiment of the present invention;

[19] Fig. 6 is an illustrative example of an agent capability file in accordance with an exemplary embodiment of the present invention; and

[20] Figs. 7a and 7b are illustrative examples of a query mapping file in accordance with an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[21] The present invention in the form of one or more exemplary embodiments will now be described. Fig. 1 is a simplified block diagram illustrating an exemplary embodiment of the present invention. Referring to Fig. 1, there is shown a system 10 representing an exemplary embodiment of the present invention. The system 10 includes an aggregation server 12, a number of agents 14, and a number of data sources 16. The data sources 16 include, for example, databases and applications which is capable of supplying data. The data sources 16 are generally organized into groups based on one or more predetermined criteria. For example, data sources 16a-c may be grouped together because these data sources 16a-c reside on a single computer system belonging to the same company. However, it should be understood that the data sources 16 need not reside on a single computer system. A person of ordinary skill in the art will know of other ways to organize a group of data sources. Moreover, each of the data sources 16 within a group may be different. For instance, within a group, one data source may be a database manufactured by one vendor, such as, IBM, and another data source may be a database from a second vendor, such as, Oracle. Each agent is designed to communicate with a specific group of data sources 16 to retrieve and integrate the desired or requested data.

[22] The system 10 generally operates in the following exemplary manner. When a user 18 wishes to retrieve certain data, the user 18 issues a request to the aggregation server 12. In an exemplary embodiment, the user 18 uses a graphical user interface on a computer to relay the request to the aggregation server 12 via a computer network 20a, such as, the

Internet. The request is encoded in XML format for delivery from the user 18 to the aggregation server 12. In an alternative embodiment, the user 18 may interact directly with the aggregation server 12 without going through any computer network.

[23] Upon receiving the request, the aggregation server 12 processes the request and determines which one or more of the agents 14 have access to the requested data. Upon identifying these agents 14, the aggregation server 12 communicates with these agents 14 via a computer network 20b to retrieve the requested data. This computer network 20b can also be, for example, the Internet. Hence, the computer networks 20a,b may or may not be the same.

[24] Each identified agent 14 further processes the request received from the aggregation server 12 and accesses the appropriate data sources 16 to retrieve the requested data. The agent 14 then integrates the retrieved data and forwards it to the aggregation server 12. The integrated data can be formatted in XML or SOAP and then forwarded to the aggregation server 12 via the computer network 20b using a number of transfer protocols including, for example, HTTP. Based on the disclosure provided herein, a person of ordinary skill in the art will know of other formats and transfer protocols which can be used to implement the data transfer between the agent 14 and the aggregation server 12.

[25] Upon receiving the retrieved data from all the relevant agents 14, the aggregation server 12 then integrates all the retrieved data and presents it to the user 18. Details with respect to how each agent 14 and the aggregation server 12 retrieves and integrates the requested data will be described further below.

[26] The operation of the system 10 is further illustrated in a more practical context as follows. In one exemplary configuration, an agent 14 resides on the private computer network of a company and is able to communicate locally with that company's internal data sources, such as, databases and applications. When a customer of the company wishes to obtain certain information relating to, for example, his/her purchase order, the customer issues a request to the aggregation server 12. The request is processed and then relayed by the aggregation server 12 to the agent 14 via, for example, the Intranet. The agent 14, in turn, retrieves the requested information from the company's data sources and then integrates the retrieved information for delivery to the aggregation server 12 which subsequently forwards the information to the customer.

[27] Fig. 2 is a flow diagram illustrating in further detail how a request issued by the user 18 is processed so as to cause one or more agents 14 to retrieve and integrate the requested data. Referring to Fig. 2, the user 18 uses a request form or a graphical user

interface to enter the request for data. The request form contains a number of different fields. Different request forms may be available to the user 18 to request different types of data. In an exemplary embodiment, the request form (and the information contained therein) is converted into an input query request encoded in XML format for delivery to the aggregation server 12. Fig. 3 shows an illustrative example of the input query request.

[28] Upon receiving the input query request from the user 18, the aggregation server 12 breaks down or converts the input query request into an internal query. In particular, for each input query request, there is a corresponding request template that can be instantiated to the internal query. The internal query is represented by a query definition file. The query definition file has two parts, namely, a head portion and a tail portion. The head portion represents the query output format, i.e., it describes what data structure is going to be displayed and the way data fusion is going to take place when data responsive to the internal query is retrieved. The tail portion represents the query input formats, i.e., it specifies what kind of data is to be retrieved and the requisite input arguments or parameters needed to retrieve such data. The tail portion is made up of a conjunctive set of query input formats. Fig. 4 provides an illustrative example of the query definition file. The purpose and use of the query definition file will be further described below.

[29] Once the internal query (and the corresponding query definition file) is created, the internal query is evaluated against a set of rules. This set of rules specifies where and how different internal queries are to be satisfied. For example, one rule may specify that a specific internal query can be satisfied by a first and a second data source; another rule may specify that this same specific internal query can also be satisfied by a third and a fourth data source. More generally, a rule can specify how a subset of the conjunctive set of query input formats in the tail portion of the internal query can be satisfied. A combined set of rules can then be used to specify how the entire internal query can be satisfied. The set of rules are designed based on the structures, constraints and contents of the data sources. In an exemplary embodiment, this set of rules is stored in a mediator specification file residing on the aggregation server 12. Each rule in the mediator specification file also includes a head portion. Similar to the tail portion of the query definition file, the head portions of the set of rules in the mediator specification file also represent the query input formats, i.e., it specifies what kind of data is to be retrieved and the requisite input arguments or parameters needed to retrieve such data. The use of this similar functionality will be further explained below. Fig. 5 provides an illustrative example of the mediator specification file.

[30] Referring back to Fig. 2, the internal query is evaluated against each rule within the set of rules in the mediator specification file. More specifically, the corresponding query definition file for the internal query is examined to determine if the tail portion of the query definition file matches a set of the head portions of rules within the mediator

5 specification file. In other words, if each of the input query formats of the tail portion of the query definition file is matched with the head portion of one rule within the set of rules, then that set of rules is considered to be a match with the internal query. It should be noted that for a rule to be considered a match with a subset of input query formats of the tail portion of the query definition file, the subset of input query formats of the tail portion of the query
10 definition file does not have to be identical to the head portion of the rule in the mediator specification file. It is sufficient to be considered a match if the subset of input query formats of the tail portion of the query definition file is equal to part or all of the head portion of the rule. In other words, the head portion of the rule may be a superset of the tail portion of the query definition file; conversely, the tail portion of the query definition file may be a superset of the head portion of the rule.

[31] For each set of matched rules, the aggregation server 12 generates a subquery. For each internal query, one or more subqueries may be generated. Each subquery identifies the data sources that are used to supply data which are responsive to the internal query as well as the agent 14 that is designed to access those identified data sources. Optionally, as
20 will be further described below, the aggregation server 12 analyzes the subqueries and formulates a query execution plan to optimize the execution of the subqueries by the relevant agents 14.

[32] For each subquery, the aggregation server 12 determines if the subquery can be executed with a set of agents. The set of agents may include one or more agents. All the
25 relevant agents 14 are then identified from the subqueries. Each agent has a corresponding agent capability file. Fig. 6 provides an illustrative example of an agent capability file. The corresponding agent capability file for each relevant agent is checked to determine if that particular agent is capable of returning the data specified by the associated subquery. A particular agent may not be able to participate in executing the associated subquery due to a
30 number of factors.

[33] If it is determined that a relevant agent is capable of participating in executing the associated subquery, the associated subquery is then formatted appropriately into an agent request and transmitted to the relevant agents for execution. In an exemplary embodiment,

the aggregation server 12 encodes the associated subquery into the agent requests in XML format and forwards it to the relevant agents via the Internet.

[34] Upon receiving the agent requests which embody the subquery, each of the relevant agents then identifies a query mapping file which corresponds to the received agent request. The query mapping file is used to map information between data in a desired format and native data retrieved from the data sources pursuant to the subquery. Furthermore, the query mapping file also includes information on how to connect to a data source thereby allowing the relevant agent to access the data source. For example, one data source to be accessed may be a database and another data source may be an application which communicates via an application programming interface. Figs. 7a and 7b provide illustrative examples of the query mapping file.

[35] For each subquery embodied in a set of agent requests, i.e., at the agent level, data which is responsive to the subquery is retrieved by the set of relevant agents from the relevant data sources. Each agent then performs join operations on the retrieved data and encodes the joined data in the appropriate format for delivery to the aggregation server 12. In an exemplary embodiment, the joined data is encoded in the XML format.

[36] Upon receiving the data from the relevant agents which executed the subqueries, the aggregation server 12 performs join, fusion and union operations on all the received data. The fusion operation aggregates data from different agents based on a set of attribute values defined in the head portion of the query definition file. The union operation unions together all aggregated data returned by the relevant agents.

[37] It should be noted that the volume of data which is responsive to the internal query may be quite high and, due to system constraints and/or other requirements, all the responsive data may not be retrieved by the relevant agents or processed by the aggregation server 12 all at once. Therefore, the amount of responsive data which is to be retrieved by the relevant agents and/or processed by the aggregation server 12 at any one time is configurable.

[38] As mentioned above, the aggregation server 12 can also analyze the subqueries and formulate a query execution plan to optimize the execution of the subqueries by the relevant agents 14. Take, for example, an internal query that has three sets of matched rules thereby generating three subqueries. The three subqueries are the same except that they each require access to different combinations of data sources. For instance, the first subquery specifies that data source A and data source B are to be accessed; the second subquery specifies that data source A and data source C are to be accessed; and the third subquery specifies that data source A and data source D are to be accessed. Without any optimization,

the agent requests corresponding to the subqueries are executed by their respective agents independently. Consequently, data source A is accessed three times.

[39] Optionally, the aggregation server 12 may optimize the execution of the subqueries as follows. First, the common data source shared by the subqueries is identified. A common key shared by all the data sources accessed by the subqueries is also identified. Then, the first subquery is executed against the common data source to retrieve, amongst other data, a list of possible values for the common key. The list of retrieved values for the common key is then concurrently passed to the relevant agents to be used to access the other data sources pursuant to the subqueries. The results retrieved pursuant to the subqueries are then unioned together to generate the final, appropriate results. This overall union operation is called star union.

[40] The foregoing optimization process is illustrated using the example given above. Furthermore, assume in the example that data source A is used to store information on component parts and their respective descriptions. The information is indexed or keyed by one field, part number. Data sources B, C and D are used to store information on component parts and their respective quantities for suppliers B, C and D. The information is indexed or keyed by one field, part number. Using the foregoing optimization process as described above, the common data source shared by the three subqueries is data source A. The common key shared by data sources A, B, C and D is the part number field. Using this information, the first subquery is executed against data source A and a list of values indexed by the part number field is retrieved. This list of values represents the part number information requested by the first subquery. This list of values is then used to retrieve the relevant information from each of the remaining data sources B, C and D pursuant to the respective subqueries. The data retrieval from the data sources B, C and D can be done in a concurrent manner. Pursuant to the first subquery (which specifies access to data sources A and B), the results retrieved from data source A are then joined with results retrieved from data source B. The joined results represent information related to selected component parts, their respective descriptions and quantities which are available from supplier B. Similarly, pursuant to the second subquery (which specifies access to data sources A and C), the results retrieved from data source A are then joined with results retrieved from data source C. The joined results represent information related to selected component parts, their respective descriptions and quantities which are available from supplier C. Likewise, pursuant to the third subquery (which specifies access to data sources A and D), the results retrieved from data source A are then joined with results retrieved from data source D. The joined results

represent information related to selected component parts, their respective descriptions and quantities which are available from supplier D. All the respective joined results are then fused together so that for each part number, the data retrieved from data sources B, C and D are aggregated together in the final results which satisfy the internal query.

5 [41] In an exemplary embodiment, the present invention is implemented in the form of control logic in either a modular or integrated manner using software. Based on the disclosure provided herein, however, it will be appreciated by a person of ordinary skill in the art that the present invention can also implemented using other methods and/or techniques, such as, hardware implementation and a combination of software and hardware
10 implementation.

[42] It is understood that the examples and embodiments described herein are for illustrative purposes only and that various modifications or changes in light thereof will be suggested to persons skilled in the art and are to be included within the spirit and purview of this application and scope of the appended claims. All publications, patents, and patent applications cited herein are hereby incorporated by reference for all purposes in their entirety.